

0MyPhoneLibrary.h 설명서

함수	설명
Address Book - MyAddressBook 클래스 [여러 주소를 관리하는 주소록에 대한 함수]	
<code>Int AddAddress(const MyAddress& address)</code>	멤버변수인 MyAddress형 배열에 address를 뒤에 덧붙인다.
<code>int DeleteAddress(size_t index)</code>	멤버변수인 MyAddress형 배열의 index번째 요소를 삭제한다.
<code>MyAddress* GetAddress(size_t index)</code>	멤버변수인 MyAddress형 배열의 index번째 요소의 주소값을 리턴한다.
<code>int UpdateAddress(size_t index)</code>	멤버변수인 MyAddress형 배열의 index번째 요소에 해당하는 '전화번호'와 '이름'을 찾아 map형인 멤버변수 m_namesByNumbers 내의 phone number 키의 value를 name으로 설정한다.
<code>Int UpdateAddress(size_t index, MyAddress& address)</code>	멤버변수인 MyAddress형 배열의 index번째 요소의 자료를 address로 바꾼다(업데이트한다).
<code>int SearchAddress(std::string name)</code>	<p>멤버변수인 MyAddress형 배열에서 name과 같은 '이름'을 갖는 요소의 인덱스를 반환한다. (조건에 맞는 요소가 없을 경우 -1을 반환)</p> <p>시간복잡도 : 평균 $O(\log n)$, 최대 $O(n)$ 시간복잡도가 최악의 경우가 나오는 n 인 경우는 Address 개수가 n인데, 모든 Address가 이름이 동일한 경우 → 이런 경우는 거의 없으므로 거의 $O(\log n)$로 작동한다.</p>
<code>int SearchAddress(const PhoneNumber& phoneNumber)</code>	<p>멤버변수인 MyAddress형 배열에서 phoneNumber와 일치하는 '전화번호'를 갖는 요소를 찾고 그 인덱스를 반환한다.(조건에 맞는 요소가 없을 경우 -1을 반환)</p> <p>처음에 번호-이름이 저장되어있는 테이블(m_namesByNumber)에서 이진탐색 후 이름을 찾고, 그 이름을 Address 배열에서 이진탐색으로 찾는다.</p> <p>시간복잡도가 최악의 경우가 나오는 n 인 경우는 Address 개수가 n인데, 모든 Address가 이름이 동일한 경우 → 이런 경우는 거의 없으므로 거의 $O(\log n)$로 작동한다.</p>
<code>size_t GetAddressCount(void)</code>	멤버변수인 MyAddress형 배열에 저장된 요소들의 개수를 반환한다.

<pre>bool CheckValidAddress(MyAddress address)</pre>	address에 기재한 그룹, 전화번호, 이름이 누락되지 않았는지 검사해 하나라도 누락되었을 시 false 반환, 그렇지 않으면 true를 반환한다.
<pre>int UpdateNumber(const PhoneNumber& number, std::string name)</pre>	Map 컨테이너 멤버변수 m_namesByNumbers에서 number 키의 value로 name을 설정한다.
<pre>int RemoveNumber(const PhoneNumber& number)</pre>	Map 컨테이너 멤버변수 m_namesByNumbers에서 number에 해당하는 요소를 삭제한다.
<pre>int AddGroup(std::string groupName)</pre>	groupName을 이름으로 하는 새로운 그룹을 생성한다.
<pre>int DeleteGroup(std::string groupName)</pre>	groupName을 이름으로 하는 그룹을 삭제한다.
<pre>std::string* GetGroup(size_t index)</pre>	string형 배열의 멤버변수인 m_myGroups의 index번째 요소의 주소값을 리턴한다.
<pre>int AddCall(Call& call)</pre>	Call형 객체인 call을 전화를 건 목록의 마지막에 덧붙인다. (시간 순으로 내림차순으로 정렬됨)
<pre>int DeleteCall(size_t index)</pre>	전화를 건 목록에서 index번째에 해당하는 기록을 지운다.
<pre>Call* GetCall(size_t index)</pre>	전화를 건 목록에서 index번째에 해당하는 기록의 call형 자료를 반환한다.
<pre>size_t GetCallCount(void) const</pre>	전화를 건 목록에 기록된 Call형 자료들의 개수를 반환한다.
<pre>int AddMessage(SMSMessage& message)</pre>	문자를 보낸 목록 마지막에 message를 덧붙이고 목록을 다시 재정렬한다. (시간 순으로 내림차순으로 정렬됨)
<pre>int DeleteMessage(size_t index)</pre>	문자를 보낸 목록의 index번째의 메시지를 지운다.
<pre>SMSMessage* GetMessage(size_t index)</pre>	문자를 보낸 목록의 index번째의 메시지를 반환한다.
<pre>int SearchFirstMessage(const PhoneNumber& number)</pre>	문자를 보낸 목록에서 number가 보낸 첫 메시지의 index를 반환한다.
<pre>size_t GetMessageCount(void) const</pre>	멤버변수 m_myMessages 내의 요소의 수를 세어 보낸 문자의 수를 반환한다.
<pre>int SortAddressByName()</pre>	'이름'의 사전 순서대로 주소록을 정렬한다.
<pre>int SortAddressByGroup()</pre>	같은 '그룹'끼리 모아 그룹 내 주소록을 사전 순서로 정렬한다. '그

	그룹들은 그룹 이름의 사전 순서대로 정렬된다.
<code>int</code> SetStringData(std::string title, std::string value)	Value를 이름으로 하는 그룹을 멤버변수 m_myGroups에 추가한다.
<code>std::string</code> GetStringData(std::string title)	모든 그룹의 이름들이 " "로 구분되어 있는 문자열을 반환한다.
<code>int</code> SaveAddressFile(std::string fileName)	fileName 파일에 멤버변수인 myAddress형 배열의 모든 주소들 (myAddress형 자료들)를 기록한다.
<code>int</code> LoadAddressFile(std::string fileName)	fileName파일에 저장된 그룹과 주소들을 불러와 그룹 이름으로 이루어진 배열과 주소들로 이루어진 배열에 저장한다.
<code>int</code> SaveCallsFile(std::string fileName)	fileName파일에 전화 기록들을 순서대로 저장한다.
<code>int</code> LoadCallsFile(std::string fileName)	fileName파일에 기록된 전화 기록(call)들을 불러와 빠른 시간 순으로 정렬한 후 배열에 저장한다.
<code>int</code> SaveMessageFile(std::string fileName)	fileName파일에 보낸 메시지들을 저장한다.
<code>int</code> LoadMessageFile(std::string fileName)	fileName파일에 저장된 메시지들을 읽어와 빠른 시간 순으로 정렬한 후 배열에 추가한다.
Address - MyAddress 클래스 [대상의 이름, 전화번호, 집주소, 사무실주소, 그룹, 생일을 관리하는 주소에 대한 함수]	
MyAddress(void)	생성자로, MyAddress형 변수를 선언할 시, 이름, 집주소, 사무실주소, 그룹, 이메일, URL을 빈 문자열로 초기화하고, 생년월일을 0으로 초기화한다. AddressID에는 프로그램 실행 후, 몇 번째로 선언된 myAddress형 변수인지가 정수형으로 기록된다.
<code>int</code> AddPhoneNumber(PhoneNumber phoneNumber)	주소에 전화번호를 뒤에 덧붙여 추가한다.
<code>int</code> ModifyPhoneNumber(size_t index, std::string phoneNumber)	주소에서 index번째로 저장되어 있는 전화번호를 phoneNumber로 바꾼다.
<code>int</code> ModifyPhoneNumber(size_t index, PhoneNumber)	

phoneNumber)	
<pre> int ModifyPhoneNumber(std::string from, std::string to) </pre>	주소에 저장되어 있는 from 전화번호를 to 전화번호로 바꾼다.
<pre> int ModifyPhoneNumber(PhoneNumber from, PhoneNumber to) </pre>	
<pre> int DeletePhoneNumber(size_t index) </pre>	주소에서 index번째에 저장된 전화번호를 지운다.
<pre> int DeletePhoneNumber(PhoneNumber phoneNumber) </pre>	주소에서 phoneNumber와 동일한 전화번호를 찾아 지운다.
<pre> PhoneNumber GetPhoneNumber(size_t index) const </pre>	주소에서 index번째 전화번호를 찾아 지운다.
<pre> size_t GetPhoneNumbersSize(void) const </pre>	주소에 저장된 전화번호의 개수를 반환한다.
<pre> int SetStringData(std::string title, std::string value) </pre>	Title에 해당하는 주소의 멤버변수에 value를 할당한다
<pre> std::string GetStringData(std::string title) const </pre>	Title에 해당하는 주소의 멤버변수의 값을 복사하여 반환한다.
<pre> int SetName(std::string name) </pre>	이름에 name을 할당한다.
<pre> std::string GetName() const </pre>	이름에 할당된 값을 반환한다.
<pre> int SetHomeAddress(std::string homeAddress) </pre>	집주소에 homeAddress를 할당한다.
<pre> int SetOfficeAddress(std::string officeAddress) </pre>	사무실주소에 officeAddress를 할당한다.
<pre> std::string GetHomeAddress(void) const </pre>	집주소에 할당된 값을 반환한다.
<pre> std::string GetOfficeAddress(void) const </pre>	사무실주소에 할당된 값을 반환한다.

<code>int</code> <code>SetInvolvedGroup(std::string groupName)</code>	그룹에 groupName을 할당한다.
<code>std::string</code> <code>GetInvolvedGroup(void)</code> <code>const</code>	그룹에 할당된 값을 반환한다.
<code>int</code> <code>SetEmail(std::string email)</code>	이메일에 email을 할당한다. 이때, email이 이메일 형식(@를 하나 포함해야 함)을 갖추지 못했을 경우 1을 반환한다.
<code>std::string</code> <code>GetEmail(void)</code> <code>const</code>	이메일에 할당된 값을 반환한다.
<code>int</code> <code>SetURL(std::string url)</code>	URL에 url을 할당한다.
<code>std::string</code> <code>MyAddress::GetURL(void)</code> <code>const</code>	URL에 할당된 값을 반환한다.
<code>int</code> <code>SetBirthDay(size_t year, size_t month, size_t day)</code>	생일을 표시하는 멤버변수에 year, month, day를 할당한다. day의 값이 월에 포함되지 않는 날짜인 경우 등 각 값들이 생년월일 형식에 맞지 않는 경우 1을 반환한다.
<code>size_t</code> <code>GetBirthDayYear(void)</code> <code>const</code>	생년월일 중 년에 해당하는 값을 반환한다.
<code>size_t</code> <code>GetBirthDayMonth(void)</code> <code>const</code>	생년월일 중 월에 해당하는 값을 반환한다.
<code>size_t</code> <code>GetBirthDayDay(void)</code> <code>const</code>	생년월일 중 일에 해당하는 값을 반환한다.
<code>int</code> <code>GetAddressID(void)</code> <code>const</code>	주소의 AddressID에 해당하는 값을 반환한다.
<code>bool operator>(const MyAddress& operand)</code>	연산자 오버로딩으로, 다른 MyAddress형 객체의 이름값(멤버변수 m_name)과의 사전식 정렬순서를 비교하는 연산자이다. 정렬순서가 뒤에 오는 것을 '이름값이 더 크다'고 표현한다.
<code>bool operator>=(const MyAddress& operand)</code>	
<code>bool operator<(const MyAddress& operand)</code>	
<code>bool operator<=(const MyAddress& operand)</code>	
<code>bool operator==(const MyAddress& operand)</code>	

<code>ostream& operator<<(ostream& out, const MyAddress& address)</code>	연산자 오버로딩으로, out 파일에 address를 기록한다.
<code>istream& operator>>(istream& in, MyAddress& address)</code>	In파일에서 데이터를 읽어 들여 address에 저장한다.
Phone number - PhoneNumber 클래스 [전화 번호를 관리하는 함수]	
<code>PhoneNumber(void)</code>	빈 문자열로 전화 번호를 초기화하는 생성자
<code>PhoneNumber(std::string phoneNumber)</code>	phoneNumber를 전화번호 형식에 맞춰 바꾼 값으로 초기화하는 생성자
<code>PhoneNumber(const PhoneNumber&)</code>	phoneNumber에 위치하는 전화번호 값으로 초기화하는 생성자
<code>Int SetPhoneNumber(std::string number)</code>	number를 전화번호 형식에 맞춰 전화번호에 값을 할당한다.
<code>std::string GetPhoneNumber(void) const</code>	전화번호로 설정된 값을 반환한다.
<code>Int Normalize(void)</code>	지역번호와 번호의 자리수에 따라 '-'을 추가하여 전화번호 형식에 맞게 설정된 전화번호를 바꾸어 준다.
<code>Bool IsEmpty(void) const</code>	전화번호에 이미 할당된 값이 있는지 확인하는 함수로, 이미 할당되어 있을 경우 false를, 할당되어 있지 않을 경우 true를 반환한다.
<code>Bool operator==(const PhoneNumber&) const</code>	다른 PhoneNumber 객체의 전화번호 값과 같은 경우 true를, 아닌 경우 false를 반환한다.
<code>Bool operator>(const PhoneNumber& operand) const</code>	해당 객체의 전화번호가 operand의 전화번호보다 사전순으로 뒤에 있을 경우 true를 반환, 아닌 경우 false를 반환한다.
<code>Bool operator<(const PhoneNumber& operand) const</code>	해당 객체의 전화번호가 operand의 전화번호보다 사전순으로 앞서 있을 경우 true를 반환, 아닌 경우 false를 반환한다.
<code>PhoneNumber& operator=(const PhoneNumber& phoneNumber)</code>	다른 객체인 phoneNumber에 할당된 전화번호 값을 전화번호에 할당한다.
<code>PhoneNumber& operator=(const std::string& phoneNumber)</code>	phoneNumber를 전화번호 형식에 맞게 바꾸어 그 값을 전화번호에 할당한다.
<code>friend std::ostream& operator<<(std::ostream&</code>	Out 파일에 전화번호인 operand를 기록한다.

out, const PhoneNumber& operand)	
friend std::istream& operator>>(std::istream& in, PhoneNumber& operand)	In 파일에서 데이터를 읽어 들여 operand에 할당한다.
Communication object - Call 클래스 & SMSMessage 클래스 [전화와 문자를 관리하는 함수]	
CommunicationObject()	생성자로, 보낸이를 빈 문자열로, 년, 월, 일, 시간, 분, 초를 0으로 초기화한다.
CommunicationObject(PhoneNumber sender, PhoneNumber receiver)	생성자로, 보낸이를 sender로, 받는이를 receiver로, 시간은 생성할 때의 시기로 초기화한다.
CommunicationObject(const CommunicationObject& object)	생성자로, Object와 동일한 값들로 초기화한다.
CommunicationObject(PhoneNumber phoneNumber, PhoneNumber receiver, int year, int month, int day, int hour, int minute, int second)	생성자로, 보낸이를 sender로, 받는이를 receiver로, 시간을 year, month, day, hour, minute, second로 초기화한다.
Int SetSender(PhoneNumber senderNumber)	보낸이로 senderNumber를 할당한다.
int SetSender(std::string senderNumber)	
PhoneNumber GetSender(void) const	메시지의 보낸이 값을 반환한다.
Int SetReceiver(PhoneNumber receiverNumber)	받는이로 receiverNumber를 할당한다.
Int SetReceiver(std::string receiverNumber)	
PhoneNumber GetReceiver(void) const	메시지의 받는이 값을 반환한다.
Void SetCurrentTime()	메시지의 시간 관련 멤버 변수들에 현재 시간을 할당한다.
Bool IsValidDate(void) const	시간 관련 멤버변수들에 할당된 값들에 문제가 없는 지를 검사하여 제대로 된 값이 할당되어 있을 시 true를, 아닐 시 false를 반환한다. Ex) m_second 멤버변수에 70이 할당되었을 경우 false를 반환

<code>Int SetTime(int year, int month, int day, int hour, int minute, int second)</code>	전달된 인자들의 값을 메시지를 보낸 시간에 할당한다.
<code>Int SetYear(int year)</code>	메시지를 보낸 년도로 year를 할당한다.
<code>Int GetYear() const</code>	메시지를 보낸 년도의 값을 반환한다.
<code>Int SetMonth(int month)</code>	메시지를 보낸 월로 month를 할당한다.
<code>Int GetMonth() const</code>	메시지를 보낸 월의 값을 반환한다.
<code>Int SetDay(int day)</code>	메시지를 보낸 날짜로 day를 할당한다.
<code>Int GetDay() const</code>	메시지를 보낸 날짜의 값을 반환한다.
<code>Int SetHour(int hour)</code>	메시지를 보낸 시간으로 hour를 할당한다.
<code>Int GetHour() const</code>	메시지를 보낸 시간의 값을 반환한다.
<code>Int SetMinute(int minute)</code>	메시지를 보낸 분으로 minute을 할당한다.
<code>Int GetMinute() const</code>	메시지를 보낸 분의 값을 반환한다.
<code>Int SetSecond(int second)</code>	메시지를 보낸 초로 second를 할당한다.
<code>Int GetSecond() const</code>	메시지를 보낸 초의 값을 반환한다.
<code>Bool operator>(const CommunicationObject& oper and)</code>	연산자 오버로딩으로, 다른 객체의 '시간'과 비교하는 연산자이다. '시간'이 더 빠를수록(할당된 값들이 작을 경우) '작다'고 표현된다. Ex) 2017년 > 2013년
<code>Bool operator>=(const CommunicationObject& oper and)</code>	
<code>Bool operator<(const CommunicationObject& oper and)</code>	
<code>Bool operator<=(const CommunicationObject& oper and)</code>	
<code>friend std::ostream& operator<<(std::ostream& out, const CommunicationObject&)</code>	Out 파일에 object를 형식에 맞추어 저장한다.
<code>friend std::istream& operator>>(std::istream& in, CommunicationObject&)</code>	In 파일으로부터 데이터를 읽어 들여 object에 저장한다.
Call 클래스	
<code>Call(PhoneNumber sender,</code>	생성자 함수이며, 보낸이에 sender값을, 받는이에 receiver값을 할

<code>PhoneNumber receiver)</code>	당하여 초기화한다.
<code>Call(const Call& object)</code>	생성자 함수이며, object와 동일한 값을 가지도록 객체로 초기화한다.
<code>Call(PhoneNumber sender, PhoneNumber receiver, int year, int month, int day, int hour, int minute, int second)</code>	생성자 함수이며, 보낸이로 sender, 받는이로 receiver, 년도로 year, 월로 month, 날짜로 day, 시간으로 hour, 분으로 minute, 초로 second를 할당하여 초기화한다.
SMSMessage 클래스	
<code>SMSMessage(PhoneNumber sender, PhoneNumber receiver)</code>	생성자 함수이며, 보낸이에 sender값을, 받는이에 receiver값을, 메시지 내용(m_message 멤버변수)에 빈 문자열을 할당하여 초기화한다.
<code>SMSMessage(PhoneNumber sender, PhoneNumber receiver, std::string message)</code>	생성자 함수이며, 보낸이에 sender값을, 받는이에 receiver값을, 메시지 내용(m_message 멤버변수)에 message 문자열을 할당하여 초기화한다.
<code>SMSMessage(const SMSMessage& object)</code>	생성자 함수이며, Object 객체와 같은 값을 가지도록 초기화한다.
<code>SMSMessage(PhoneNumber sender, PhoneNumber receiver, std::string message, int year, int month, int day, int hour, int minute, int second)</code>	생성자 함수이며, 보낸이로 sender, 받는이로 receiver, 메시지 내용(m_message 멤버변수)으로 message 문자열, 년도로 year, 월로 month, 날짜로 day, 시간으로 hour, 분으로 minute, 초로 second를 할당하여 초기화한다.
<code>Int SetMessage(std::string message)</code>	메시지 내용(m_message 멤버변수)에 message 문자열을 할당한다.
<code>std::string GetMessage(void) const</code>	메시지에 할당된 메시지 내용(m_message 멤버변수) 문자열을 반환한다.
<code>friend std::ostream& operator<<(std::ostream& out, const SMSMessage& object)</code>	In 파일에서 데이터를 읽어 들여 object에 값들을 할당한다.
<code>friend std::istream& operator>>(std::istream& in, SMSMessage& object)</code>	Out 파일에 object를 메시지 형식에 맞추어 저장한다.
<code>Bool operator>(const SMSMessage& operand)</code>	연산자 오버로딩으로, 연산자 대상인 operand의 보낸이(멤버변수 m_sender)를 번호크기를 비교한다. 전화번호의 왼쪽 자리의 수부터 비교한다. 두 객체의 전화번호가 같을 때는 메시지를 보낸 시간을 비교한다. (시간이 더 빠를수록 '작다'고 표현한다)
<code>Bool operator>=(const SMSMessage& operand)</code>	
<code>Bool operator<(const</code>	

SMSMessage& operand)	Ex) 010-0111-1111 < 010-1111-1111
Bool operator<=(const SMSMessage& operand)	Ex) 2017년 > 2013년